Doc Code: AP.PRE.REQ

PRE-APPEAL BRIEF REQUEST FOR REVIEW	Docket Number (Optional) SCS-550-541	RECEIVED CENTRAL FAX CENTER

	Application Number 10/807,498	Filed March 24, 2004	MAR 09 2007

First Named Inventor BUTCHER		

Art Unit 2183	Examiner A. Li

Applicant requests review of the final rejection in the above-identified application. No amendments are being filed with this request.

This request is being filed with a notice of appeal.

The review is requested for the reason(s) stated on the attached sheet(s).
        Note: No more than five (5) pages may be provided.

I am the

☐ Applicant/Inventor

☐ Assignee of record of the entire interest. See 37
    C.F.R. § 3.71. Statement under 37 C.F.R. § 3.73(b)
    is enclosed. (Form PTO/SB/96)

☒ Attorney or agent of record   _____27,393_____
                                    (Reg. No.)

☐ Attorney or agent acting under 37CFR 1.34.
    Registration number if acting under 37 C.F.R. § 1,34 _____

_____
Signature

Stanley C. Spooner

_____
Typed or printed name

_____703-816-4028_____
Requester's telephone number

_____March 9, 2007_____
Date

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below.*

☒ *Total of 1 form/s are submitted.

STATEMENT OF ARGUMENTS IN SUPPORT OF
PRE-APPEAL BRIEF REQUEST FOR REVIEW

The following listing of clear errors in the Examiner's rejection and his failure to identify

essential elements necessary for a *prima facie* basis of rejection is responsive to the Final Rejection

mailed December 13, 2006 (Paper No. 20061206).

Error #1. The Examiner misstates the language of claims
31-60 in contending that they are indefinite

The Examiner rejects claims 31-60 under 35 USC §112 as being indefinite, stating that it is

"unclear how a computer program product **contains** a computer readable storage medium" (emphasis

added). The actual claim language in the independent computer program products claims 31 and 46 states

"a computer program product **comprising** a computer readable storage medium . . . ." Thus, Applicants'

claims accurately state that the computer program product includes, but is not limited to, a "computer

readable storage medium" which limitations meet the Patent Office imposed requirement for a "tangible

result" with respect to computer program product claims. The Examiner simply misstates Applicants'

claim language which has, as previously noted in the last Amendment (page 18), been accepted by the

Patent Office in U.S. Patent 6,836,860 and therefore does not evidence any indefiniteness.

Error #2. The Examiner's conclusion in section 10 of the Final Rejection is
inconsistent with the admission made in section 11

In the outstanding Final Rejection, the Examiner admits that "Click has not explicitly taught . .

." reciting the two structural elements recited in independent claim 1 (and repeated in the other

independent claims 16, 31 and 46). While this admission is very much appreciated, the Examiner then

suggests that the interrelationship between these elements (set forth in claim 1 after the "wherein"

clause at line 5) is somehow disclosed in the Click reference. Simple logic dictates that, if the

structures are not taught in Click, the claimed interrelationship between those unclaimed structures

cannot possibly be taught.

Appl. No. 10/807,498
March 9, 2007

In view of the Examiner's admissions, Click cannot possibly teach the claimed interrelationship between the structures given that the structures aren't present in Click. Accordingly, the rejection of independent claims 1, 16, 31, 46 and all claims dependent thereon over the Click/Smith combination clearly falls in view of the Examiner's admissions.

Error #3. The Examiner has failed to understand that the operation of the Click and Smith references are mutually incompatible

The Examiner alleges that it would be obvious to "incorporate the processor of Smith in the device of Click to increase processor performance and capability." However, if one of ordinary skill in the art followed the teachings of Smith and did not use a translating program, then he would not be able to identify the redundant null checks in advance and remove them as is taught (and required) by the Click reference. Conversely, using a separate translating program such as disclosed in Click removes the need for any hardware-based conversion, such as Smith.

The Examiner's assertion that one of ordinary skill in the art would be motivated to combine Click and Smith for the generic benefits of "increase[d] processor performance and capability" fails to address the basic reality that the teaching of these two documents is fundamentally incompatible and contradictory. The Click and Smith references are mutually incompatible and therefore cannot and would not be combined as suggested by the Examiner.

Error #4. The Examiner's attempt to respond to the previous Amendment is completely unsupported

In section 38 of the Final Rejection, the Examiner attempts to respond to the issues with respect to the combination of the Click and Smith references contained in Applicants' previously filed Amendment. The Examiner starts with the unsupported allegation that "the decoder in the claim language is found within the combined references." Of course, this is inconsistent with the Examiner's admission in section 11 that Click does not explicitly teach the claimed "instruction decoder." If disclosure of the decoder, as the Examiner admits, is not present in the Click reference, then it must be

- 2 -

Appl. No. 10/807,498
March 9, 2007

disclosed only in the Smith reference. Therefore, the Examiner's statement that the decoder "is found within the combined references" is clearly in error as, at best, it is in only one reference, Smith.

In lines 2-5 of section 38, the Examiner offers a partial quotation of the language of claim 1. However, the Examiner omits the portion of the claim language which specifies that the instruction decoder is "in response to a memory access instruction" and that the instruction decoder "compares a base register value . . . with a predetermined null value" and then, as appropriate, "triggers branching." The Examiner is simply ignoring portions of claim 1 in an attempt to show that the structures are present in Click, even though the Examiner admits that those claimed structures are missing from the Click reference. The Examiner should appreciate that, in Click, it is not the decoder that performs null value checking. Rather, in Click it is some other processing logic controlled by specifically provided null value checking program instructions.

Thus, the Examiner's rationale in section 38 is unsupported by the references and is contrary to the limitations set out in Applicants' independent claims. Thus, the Examiner has not rebutted Applicants' indication that there is no motivation for combining the Click and Smith references under the provisions of 35 USC §103. While the Examiner has provided substantial verbiage, he has not provided any real "reason" or "motivation" and indeed has ignored the fact that both references would lead one of ordinary skill in the art away from the claimed invention.

Error #5. The Examiner erroneously resorts to an ambiguous reference to "open claim language" in order to try to support his argument that Click, which admittedly doesn't teach the claimed structures, somehow teaches the claimed interrelationship between structures

In section 38, and in particular the portion bridging pages 13 and 14 of the Final Rejection, the Examiner argues that Applicants' claims somehow have "open claim language" and that "the decoder can receive other instructions to perform the null exception handling." Applicants' review of the claim language specifies and recites in accordance with U.S. patent practice not only the structures, but the interrelationship between structures which must be present in a cited prior art reference in order to

- 3 -

Appl. No. 10/807,498
March 9, 2007

support a rejection under §§102 or 103. The past admission that the structures are missing from Click

is appreciated. The Examiner's attempt to assert "open claim language" is an attempt to avoid the

burden of showing that somewhere in the two cited prior art references there must be some disclosure

of Applicants' claimed elements and interrelationship between elements.

The Examiner also attempts to suggest that the Click reference does its null exception handling

in response to a memory access instruction. This is simply incorrect. In Click, program instructions,

which are not memory access instructions, have to be provided within the program or the JVM to

check a pointer before it is used to determine whether or not it is the null value. This takes place

before any memory access instruction is decoded and executed. Therefore, it cannot operate "in

response to a memory access instruction" as required by Applicants' claim.

Error #6. The Examiner erroneously references Tanenbaum which is not applied in any pending rejection

The Examiner makes the incredible statement that "whether the functionality is done in

hardware or software is irrelevant." The Examiner is clearly not well versed in the data processing

apparatus and data processing systems. In the real world, it makes a great deal of difference whether a

function is performed by hardware or software. There are massive differences in the performance,

energy consumption, complexity and many other issues associated with whether a function is

performed in software or hardware.

The basic principle underlying Java as used in the Click reference is to abstract the processing

away from any dependence upon hardware support for particular functionality. In this context, the real

world very particular reasons are needed to build back in a dependence upon supported hardware

functionality. The present invention identifies the null value checking as a particular piece of

functionality which can readily be supported by hardware and yields a substantial efficiency gain in

order to justify its provision.

- 4 -

Appl. No. 10/807,498
March 9, 2007

The Examiner's conclusion that "whether functionality is done in hardware or software is irrelevant" is clearly the Examiner's own personal opinion, is not the view of those having even ordinary skill in the data processing art as noted above and is not the basis be evaluation under 35 USC §103. It is unclear as to whether the Examiner is attempting to combine the Tanenbaum reference with the Click and Smith combination or other combinations. However, since this reference has not yet been applied as any statutory basis, it is not of record in the pending rejections.

Error #7. The Examiner fails to appreciate that the Click reference would lead one of ordinary skill in the art away from the claimed invention

Because the Examiner admits that it fails to teach the structure and yet the Examiner contends it teaches the structural interrelationship, Click would appear to lead one of ordinary skill in the art away from the structures recited in Applicant' claims. The Examiner's citation of Tanenbaum, as questioned above, is unclear, but assuming it is to be combined with Click and Smith, clearly recognizes that the decision regarding whether functionality is to be executed in hardware or software has a profound influence upon real-life performance. Tanenbaum specifically states at page 11, paragraph 4, "the decision to put certain functions in hardware and others in software is based upon such factors as cost, speed, reliability, and frequency of expected changes."

The Click reference, as the Examiner contends, teaches that less null checks means smaller and faster code so that one should remove null checks. Applicants admit that this was common knowledge that pre-dates even the Click reference. However, Applicants' claimed invention does not specifically remove null checks and, because of the way the claimed invention works, Applicants actually add more. Every load and store in the instruction sets performs a null check. Given the Click teaching, the claimed invention should not actually provide an advantage. Yet, when implemented in hardware, it does, although it is likely that if implemented in software, it would not. Click and its teachings would clearly lead one of ordinary skill in the art away from utilizing hardware for null checks as set out in Applicants' claims.

- 5 -